

**WHAT IS CLAIMED IS:**

1. A method of servicing a client request in a server comprising:  
determining a first service interface subclass corresponding to the client  
request from one or more service interface subclasses derived from a  
request interface base abstract class; and  
servicing the client request using the first service interface subclass, wherein  
the first service interface subclass corresponds to at least one service  
provided by the server at a corresponding port.
2. A method according to claim 1, wherein the one or more service  
interface subclasses comprise at least one protocol handler for servicing the client  
request at the corresponding port.
3. A method according to claim 1, wherein the first service interface  
subclass is determined from a configuration file.
4. A method according to claim 3, wherein the configuration file  
comprises descriptions of  
the one or more services provided by the server; and  
corresponding ports for the one or more services.
5. A method according to claim 1, wherein one of the one or more service  
interface subclasses is configured to service Post Office Protocol requests at the  
corresponding port.
6. A method according to claim 1, wherein one of the one or more service  
interface subclasses is configured to service Simple Mail Transfer Protocol requests at  
the corresponding port.
7. A method according to claim 1, wherein a parent process performs the  
determining and servicing.
8. A method according to claim 7, further comprising:

creating one or more child processes, wherein the one or more child processes are configured to service client requests.

9. A method according to claim 8, further comprising:  
creating one or more execution threads for each child process.

10. A method according to claim 9, further comprising:  
selecting a socket requested by the client request;  
blocking the selected socket; and  
upon servicing the client request, unblocking the selected socket.

11. A method according to claim 1, wherein the at least one service includes a scanning service.

12. A method according to claim 8, further comprising:  
adjusting a number of child processes according to a load of the server.

13. A server communication channel architecture for servicing at least one client request on one or more socket ports on a server, the service communication channel architecture comprising:

an abstract base class describing one or more service interfaces, wherein each service interface is configured to service at least one client connection corresponding to the at least one client request on the one or more socket ports; and

a process pool of one or more child processes, wherein each child process is configured to create a pool of one or more execution threads configured to call the one or more service interfaces of the abstract base class for servicing the at least one client request.

14. A server communication channel architecture according to claim 13, further comprising:

one or more subclasses derived from the abstract base class and corresponding to the one or more service interfaces, wherein the one or more

subclasses represent at least one service provided by the server at a particular socket port.

15. A server communication channel architecture according to claim 13, further comprising:

at least one parent process configured to  
create at least one socket for the at least one client connection;  
create the one or more service interfaces of the abstract base class;  
create the process pool of one or more child processes; and  
monitor the one or more child processes.

16. A server communication channel architecture according to claim 15, further comprising:

at least one configuration file, wherein the at least one configuration file  
describes one or more services provided by the server and one or more  
ports corresponding to the one or more services.

17. A server communication channel architecture according to claim 16, wherein the parent process is further configured to use the at least one configuration file to create the one or more service interfaces of the abstract base class.

18. A server communication channel architecture according to claim 16 wherein the one or more child processes are further configured to  
determine a number of client requests that can be serviced by each child  
process; and  
determine a number of execution threads to be created by each child process.

19. A server communication channel architecture according to claim 13, wherein the execution threads are configured for serial access to the one or more socket ports.

20. A server communication channel architecture according to claim 13, wherein one of the one or more service interfaces is configured to service Post Office Protocol requests.

21. A server communication channel architecture according to claim 13, wherein one of the one or more service interfaces is configured to service Simple Mail Transfer Protocol requests.

22. A server communication channel architecture according to claim 14, wherein the at least one service provided by the server includes a scanning service.

23. A server communication channel architecture of claim 13, wherein a pointer is used to indicate a particular service interface corresponding to a particular socket port of the client connection.

24. A server communication channel architecture of claim 13, wherein each execution thread is further configured to  
prior to calling the one or more service interfaces, determine one of the one or  
more socket ports corresponding to the at least one client request; and  
determine one of the one or more service interfaces corresponding to the client  
request.